

**WHAT IS CLAIMED IS:**

1 Sub  
2 B2 > 1. A method of aligning instructions in a  
processor comprising:  
  
3 aligning a first instruction;  
  
4 decoding the size of the first instruction;  
  
5 determining the beginning of a second instruction  
6 based on the size of the first instruction;  
  
decoding the size of the second instruction;  
  
determining whether processing the second  
instruction will deplete one of a plurality of buffers; and  
  
instructing the one of the plurality of buffers to  
receive additional data if processing the second instruction  
depletes the one of the plurality of buffers.

1 2 The method of Claim 1, further comprising  
2 storing the plurality of instructions in a plurality of sub-  
3 buffers.

1 3. The method of Claim 1, further comprising  
2 comparing a most significant bit of a pointer to a first of  
3 the plurality of sub-buffers to a most significant bit of a  
4 pointer to a second of the plurality of sub-buffers to

5 determine whether processing one of the plurality of  
B2 instructions will deplete a buffer.

1           4. The method of Claim 1, further comprising  
2 storing a first instruction across a plurality of storage  
3 elements prior to processing the instructions.

1           5. The method of Claim 1, further comprising  
2 adding the size of the first instruction to a current  
3 instruction position to determine the beginning of the second  
4 instruction.

5           6. The method of Claim 1, further comprising  
aligning ahead a number of cycles equal to a cache latency.

6           7. The method of Claim 1, further comprising  
aligning instructions in a digital signal processor.

7           8. The method of Claim 1, further comprising  
issuing a request to a memory to reload the plurality of  
8 buffers.

9           9. A method of processing instructions within a  
processor comprising:

10           predicting whether one of a plurality of buffers  
11 will be depleted of instruction data within a number of cycles  
12 approximately equal to a cache latency;

6 preparing the one of a plurality of buffers to be  
7 loaded with additional instruction data if the one of the  
8 plurality of buffers will be depleted.

1 10. The method of Claim 9, further comprising  
2 decoding the size of the a first instruction in the  
3 instruction data;

4 determining the beginning of a second instruction in  
5 the instruction data based on the size and position of the  
6 first instruction; and

7 decoding the size of the second instruction.

11. The method of Claim 9, wherein the plurality of  
buffers are divided into a plurality of sub-buffers.

12. The method of Claim 11, wherein the predicting  
is accomplished by comparing a most significant bit of a  
pointer to a first of the plurality of sub-buffers to a most  
significant bit of a pointer to a second of the plurality of  
sub-buffers to determine whether processing one of the  
plurality of instructions will deplete one of a plurality of  
buffers.

13. The method of Claim 9, further comprising  
2 aligning the instruction data.

<sup>1</sup> B2  
1 14. The method of Claim 9, further comprising  
2 processing the instructions in a digital signal processor.

1 15. The method of Claim 9, further comprising  
2 issuing a request to reload the plurality of buffers.

1 16. A processor comprising:  
2  
3 a plurality of buffers adapted to store first  
instruction data including a plurality of instructions;  
4  
5 an instruction request unit adapted to align the  
plurality of instructions for execution;  
6  
7 a width decoder adapted to determine the size of the  
plurality of instructions;  
8  
9 a transition detector adapted to predict when one of  
the plurality of buffers will be empty, the transition  
detector adapted to send a signal to instruct one of the  
11 plurality of buffers to load a second instruction data.

1 17. The processor of Claim 16, wherein the  
2 plurality of buffers is divided into a plurality of sub-  
3 buffers.

1 18. The processor of Claim 16, wherein the  
2 transition detector compares a most significant bit of a  
3 pointer to a first of the plurality of sub-buffers to a most

4 B2 significant bit of a pointer of a second of the plurality of  
5 sub-buffers to determine whether processing one of the  
6 plurality of instructions will deplete a buffer.

1 19. The processor of Claim 16, wherein the  
2 processor aligns ahead a number of cycles equal to a cache  
3 latency.

1 20. The processor of Claim 16, wherein the  
2 processor is a digital signal processor.

3 21. An apparatus, including instructions residing  
4 on a machine-readable storage medium, for use in a machine  
5 system to align instructions in a processor, the instructions  
6 causing the machine to:

7 receive data containing instructions in a plurality  
8 of buffers;

9 decode the size of a first instruction;

10 determine the beginning of a second instruction  
11 based on the size of the first instruction;

12 decode the size of the second instruction;

determine whether processing the second instruction  
will deplete one of the plurality of buffers; and

13                   instruct the one of the plurality of buffers to  
14 B2   receive additional data if processing the second instruction  
15                   depletes the one of the plurality of buffers.

1                   22. The apparatus of Claim 21, wherein the  
2                   plurality of instructions are stored in a plurality of sub-  
3                   buffers.

1                   23. The apparatus of Claim 21, wherein a most  
2                   significant bit of a pointer to a first of the plurality of  
3                   sub-buffers is compared to a most significant bit of pointer  
                      to a second of the plurality of sub-buffers to determine  
                      whether processing one of the plurality of instructions will  
                      deplete a buffer.

1                   24. The apparatus of Claim 21, wherein a first  
2                   instruction is stored across a plurality of storage elements  
                      prior to processing the instructions.

Add  
B2